IJIST
P ISSN : 2618-1630
E ISSN : 2909-6130

A Comparative Analysis of Camera, LiDAR and Fusion Based Deep Neural Networks for Vehicle Detection

Shafaq Sajjad[1], Ali Abdullah[1], Mishal Arif[1], Muhammad Usama Faisal[1], Muhammad Danish Ashraf[2], Shahzor Ahmad[1]

[1]College of Electrical & Mechanical Engineering, NUST.

[2]Synapsify Systems.

* Correspondence: Shafaq Sajjad,shafaqsajjad95@gmail.com

Citation | Sajjad. S, Abdullah. A, Arif. M, Faisal. M. U, Ashraf. M. D, Ahmad. S, "A Comparative Analysis of Camera, LiDAR and Fusion Based Deep Neural Networks for Vehicle Detection". International Journal of Innovations in Science and Technology. Vol 3, Special Issue, pp: 177-186, 2021.

Received | Nov 15, 2021; Revised | Jan 27, 2022 Accepted | Jan 28, 2022; Published | Jan 29, 2022.

Self-driving cars are an active area of interdisciplinary research spanning Artificial Intelligence (AI), Internet of Things (IoT), embedded systems, and control engineering. One crucial component needed in ensuring autonomous navigation is to accurately detect vehicles, pedestrians, or other obstacles on the road and ascertain their distance from the self-driving vehicle. The primary algorithms employed for this purpose involve the use of cameras and Light Detection and Ranging (LiDAR) data. Another category of algorithms consists of a fusion between these two sensor data. Sensor fusion networks take input as 2D camera images and LiDAR point clouds to output 3D bounding boxes as detection results. In this paper, we experimentally evaluate the performance of three object detection methods based on the input data type. We offer a comparison of three object detection networks by considering the following metrics - accuracy, performance in occluded environment, and computational complexity. YOLOv3, BEV network, and Point Fusion were trained and tested on the KITTI benchmark dataset. The performance of a sensor fusion network was shown to be superior to single-input networks.

Keywords: Sensor fusion; object detection; 3D object detection; LiDAR point cloud; self-driving cars.

Introduction.

Object detection has taken primary importance in autonomous driving. At present, current perception systems utilize input data in the form of 2D images, point clouds, or a combination of both 2D images and LiDAR point clouds to achieve accurate 3D localization and detection of vehicles. Research in the field of object detection has produced mature algorithms for 2D images. The emergence of Region-based Convolutional Neural Network (RCNN) [1], Fast-RCNN [2], and Faster-RCNN

[3] removed the bottlenecks of large operating time and high computation power in 2D object detection. Different from region-based algorithms, multiple versions of the state-of-the-art object detector You Only Look Once (YOLO) [5, 6,7] have eased the task of predicting bounding boxes and class probabilities in 2D images.

LiDAR is a widely used sensor in obtaining distances between the object and the sensor. LiDAR emits an infrared laser beam to determine the distance via the time-of-flight principle. The wavelength of LiDARs exploited in self-driving cars is in class 1 eye-safe range. In general, LiDARs perform comparatively better in challenging weather conditions such as fog and rain as opposed to optical cameras. These sensors are also relatively more resilient to changes in ambient light conditions. While 2D LiDARs featuring an array of beams such as those from Sick or LeddarTech are typically manipulated in Intelligent Transportation Systems (ITS), self-driving cars make use of 3D LiDARs such as those from Ouster or Velodyne. These 3D LiDARs maneuver a rotating swivel that covers the entire field of view by scanning an array of laser beams across it. The infrared lasers are in the form of pulses and objects reflect these pulses hence distance information is obtained, yielding a 3D point cloud of the surrounding environment. Vertical resolution and angular revolution are key features that dictate the choice of a 3D LiDAR in an application. Currently, the common use of LiDARs is limited by their high cost.

For the challenge of 3D object detection using LiDAR point clouds, the computer vision community has developed several methods. These include point-cloud voxels [2,3], or transforming the 3D view of point-cloud into a top-down 2D view to exhibit objects [1]. Some other techniques focus on estimating the 6-DoF from a sequence of images. Point Net [9] architecture has garnered significant attention in the research cadre of autonomous vehicles. A variant of this architecture has been applied in Point Fusion [14] to devise an application-agnostic algorithm. However, point clouds do not output color information and, depending upon the resolution of the sensor, point clouds are more or less sparse [15].

Another approach to 3D object detection combines LiDAR data and 2D images. This method has been manifested to benefit from the complementary capabilities of cameras and LiDARs. In a conceptually simplistic approach, a 2D detection network has been utilized to make detections in the 3D point cloud [15]. This is achieved by fusing output from a 2D detector with a transformed 3D point cloud. More involved approaches include Point Fusion [14] and Multi-View 3D Network (MV3D) [12] where a region-based fusion approach has been proposed.

In this paper, we classified object detection algorithms based on input data. Three object detection networks have been identified in this regard: 1) YOLO v3 [7] for 2D images, 2) BEV detection for point clouds, and 3) Point Fusion [14] for fused data. We retrained these algorithms on KITTI [8] and specified metrics to assess the performance of these networks. The same dataset was put into service to evaluate all three networks to ensure the completeness of the comparison. Metrics for performance evaluation are accuracy, performance in occluded environment, and computational complexity.

Following are the main contributions of this work:

•We surveyed existing object detection methods for 2D images, point clouds, and sensor fusion networks.

•We chose representative methods in each category and retrained them on the KITTI dataset [8], and evaluate their performance based on accuracy, performance in occluded environment, and computational complexity.

•We conveyed an analysis of the results along with the advantages and disadvantages of each algorithm.

Related Work.

This section highlights outstanding object detection works for 2D images, point clouds, and sensor fusion. It also reviews the performance of each network in comparison with other networks.

2D Image Approaches

Object detection networks detect certain object classes within an image. Two main categories of state-of-the-art methods can be identified: one-stage methods and two stage-methods. YOLO [5], RetinaNet [18], and Single Shot Multibox Detector (SSD) [19] are one-stage methods that prioritize inference speed. On the other hand, detection accuracy takes precedence in two-stage networks as they first propose candidate regions having a high likelihood of staging the objects, and then score these regions to provide the final detections. Examples include Faster R-CNN [3] and Mask R-CNN [4]. The task of bounding box estimation has great importance in the object detection problem. In some previous works, box encoding is applied where center coordinates (x,y) and offset of the bounding box are considered. RCNN [1], Fast RCNN [2], Faster-RCNN [3], YOLO [5], YOLOv2 [6], and Mask R-CNN [4] wield this type of encoding method for bounding box with a slightly different loss calculation scheme. YOLO [5] bases detection on a regression model. Image is divided into a grid of size S*S and B number of bounding boxes, their confidence scores, and class probabilities are predicted for each cell. Predictions are then encoded as a tensor. As compared to RCNN [1], YOLO [5] offers a faster detection speed. However, there is a slight reduction in performance.

Point cloud Approaches

3D Fully Convolutional Network (FCN) extended the application of 2D FCN by applying it to point cloud data [13]. In some previous works, sophisticated segmentation algorithms have been applied to propose candidates [17]. Region Proposal Network (RPN) is a more recent method of candidate proposal. Complex-YOLO [10] proposed Euler-Region-Proposal Network (E-RPN) for pose estimation. Moreover, PointNet [9] is a ground-breaking contribution that consumes raw point cloud data and is compatible with several applications including part segmentation, object classification, and detection. Some other object detection algorithms transform point clouds into multiple views including Front View and Bird's Eye View (BEV) multi-view feature maps [12,15]. A similar approach is adopted by VeloFCN [16], where point-cloud is transformed into front view. Our algorithm also takes advantage of BEV transformation to perform object detection in point clouds.

Sensor Fusion Approaches

MV3D [12] is a sensor fusion network that efficiently deals with the limitations associated with the sparse nature of point clouds. It transforms point clouds into multiple views to make accurate 3D predictions. Moreover, the network is conveniently divided into two sub-networks: the first sub-network generates 3D candidate box proposals and the second sub-network fuses features from multiple modalities. This fusion framework rejects redundant features. A more conceptually simplistic approach is provided in [15] where 2D detections from a CNN are projected onto the 3D point cloud to obtain LiDAR point subset. A novel model-fitting algorithm then identifies the 3D bounding box based on generalized car models. Point Fusion [14] is a more recent contribution in 3D object detection that processes 3D point cloud data and 2D image data separately with PointNet [9] architecture and a CNN respectively. Information loss associated with BEV point clouds is mitigated in this algorithm.

Material and Methods.

In this paper, we focused on the car detection problem. KITTI [8] benchmark dataset was employed to retrain open-sourced algorithms. The 3D object detection task of the KITTI [8] dataset contains aligned 2D images and point clouds. Labels were available in the form of 2D and 3D bounding boxes. A total of 7418 point clouds and corresponding 2D images were adopted for training the networks from the KITTI [8] benchmark dataset. For 2D images, YOLOv3 [7] was evaluated and for point cloud data, a 2D projection approach to Bird's Eye View (BEV) was adapted as arrayed in MV3D [12] and 3D FCN [17]. Point Fusion [14] was retrained and evaluated as a framework for sensor fusion.

KITTI [8] dataset was obtained via VM Station Wagon mounted with number of different sensors including Velodyne HDL64 high precision Global Positioning System (GPS) inertial navigation system and RGB camera. A total of 6-hour drive data was obtained from driver viewpoint in [8]. Velodyne HDL64 rotates at 10Hz frequency with angular resolution of 0.09°. It captures 1.3 million points points/second with 360° horizontal and 26.8°vertical field of view having range of 120m [8]. In this paper, we have focused on left camera RGB images, corresponding point clouds, and calibration files including the calibration details for velodyne to camera calibration.

Following method was adopted to implement and evaluate the performance of three object detection algorithms that are, YOLOv3, BEV network, and Point Fusion:

i. KITTI dataset was obtained from its website as it is an open-source dataset.

ii. The training dataset was split into training and validation dataset in the ratio 1:1. It was used to categorize three classes of objects namely car, pedestrian and cyclist.

iii. YOLOv3 was trained using RGB images only as it is a 2D detection network and does not require point cloud data.

iv. BEV network was trained using point clouds only as this framework makes detections in point clouds. Point clouds were projected into BEV to encode the information of density height and intensity. Firstly, the height feature was obtained by discretizing the point cloud into a 2D grid with a 0.1m resolution. Secondly, in every cell, the value of reflectance of every point having maximum height was obtained. Thirdly, the density feature simply proclaimed the total number of points in a cell. By implementing these steps, BEV portrayal of point clouds was obtained for the dataset.

v. Point Fusion was trained using both RGB images and LiDAR point clouds as it is a sensor fusion network.

vi. These trained frameworks were then tested using test images available in the dataset.

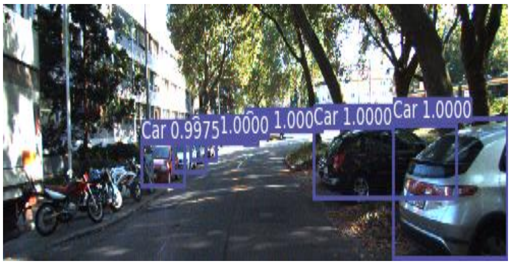
Results and Discussion.

Table 1.shows a comprehensive comparison of models trained and evaluated on the KITTI benchmark dataset for car detection.

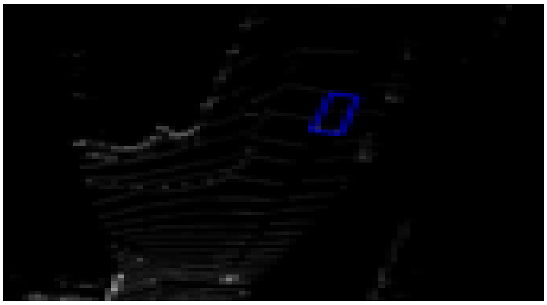
Table 1. Comparison of Object Detection Networks trained on KITTI

Method	**Type of Input Data**	**Input Processing**	**No. of Stages**	**Average Precision (%)**	**Inference Time (sec)**
YOLO v3	2D images	S*S grid	1	45	0.5
BEV	Point-clouds	2D projection	2	42	0.9
Point Fusion	2D images + point clouds	PointNet + ResNet	2	47.8	1.2

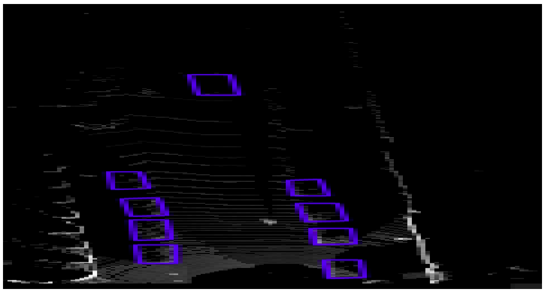
(a) YOLOv3 Detection Result



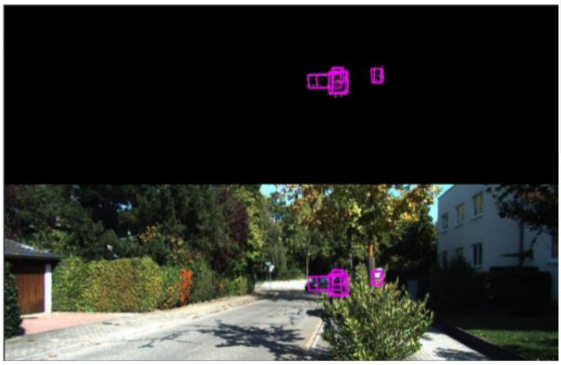
(a) YOLOv3 Detection Result



(b) BEV Detection Result

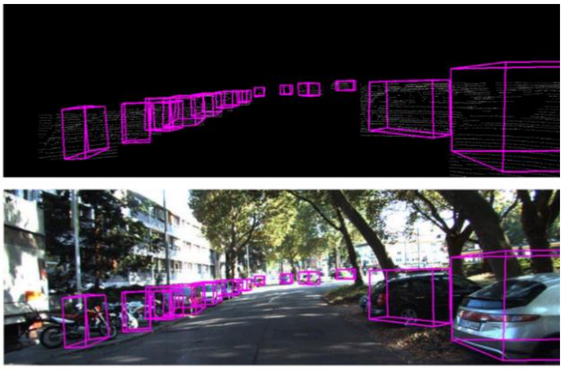


(b) BEV Detection Result



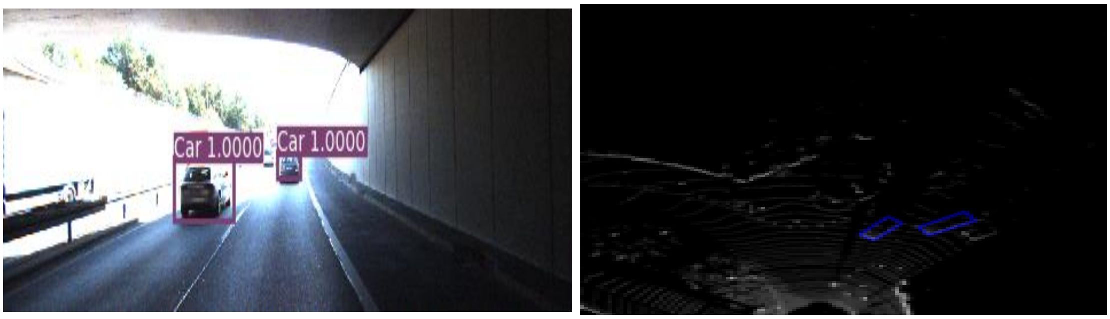
(c) Point Fusion Detection Result

Figure 1.Object Detection Results for Three Implemented Algorithms. (a) YOLOv3: Partially Visible Vehicles not detected (b) BEV: Partially Visible Vehicles detected (c) Point Fusion: Partially Visible Vehicles detected



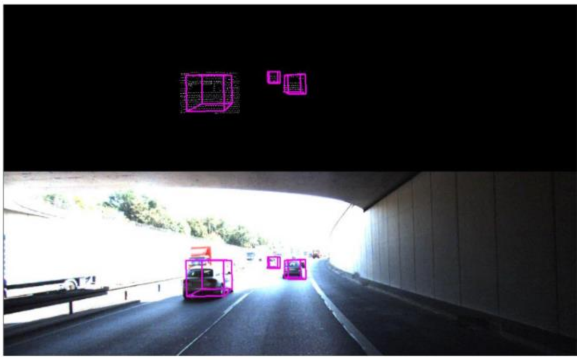
(c) Point Fusion Detection Result

Figure 2. Object Detection Results for Three Implemented Algorithms. (a) YOLOv3: Missed Detections in Occluded Environment (b) BEV: Occluded Vehicles Detected (c) Point Fusion: Maximum Number of Occluded Vehicles Detected



(a) YOLOv3 [7] Detection Result

(b) BEV Detection Result



(c) Point Fusion [14] Detection Result

Figure 3. Object Detection Results for Three Implemented Algorithms. (a)YOLOv3: Detections Missed (b) BEV: Detections Missed (c)PointFusion: No Missed Detections

Discussion.

Three metrics were chosen to evaluate the performance of each algorithm: 1) Accuracy, 2) Performance in Occluded Environment, 3) Computational Complexity.

Accuracy

Average precision (AP) was considered as a metric to determine the accuracy of detections. AP scores for implemented algorithms are listed in Table 1. As compared to YOLOv3 and BEV, Point Fusion gives higher AP. Projection losses associated with BEV reduced detection accuracy. On the other hand, YOLOv3 displayed reduced performance with KITTI dataset. Qualitative results depicted in Figure 1-3 revealed that sensor fusion gives the best detection results in all scenarios. Hence, the sensor fusion network proposed in Point Fusion overcomes the drawbacks associated with single sensor networks.

Performance in Occluded Environment

The closeness or merging of two factors such that one is completely or partially covered by the other is referred to as occlusion. Object detection in an obstructed environment is a beneficial indicator of algorithm performance due to the problem's complexity. As seen in Fig. 2, YOLOv3 and BEV missed most occluded objects whereas the sensor fusion network detected all occluded objects available in the scene. From the results delineated in Figure 1-3, it was observed that YOLOv3 gives poor performance, BEV gives an intermediate performance, and Point Fusion produces the most accurate results in an occluded environment. This proves that sensor fusion frameworks are suited for application in all types of scenarios.

Computational Complexity

YOLOv3 has a fully convolutional architecture comprised of 106 layers. It is the slowest network compared to BEV and Point Fusion; however, it is less sophisticated than many other detection networks. On the other hand, BEV is the least complex algorithm as it projected a 3D point cloud into a 2D point cloud using the method offered in MV3D and made detections using Faster-RCNN. Point Fusion lies between the other two algorithms in terms of computational complexity. Moreover, the performance of Point Fusion was increased by the adoption of PointNet that processed point clouds in raw form. From qualitative results subdued in Figure 1-3 and AP unveiled in Table 1, it can be derived that the increased computational complexity of sensor fusion frameworks can be overlooked owing to their increased detection accuracy.

In Figure 1-3 qualitative results are set forth. In Point Fusion detection results, front views of corresponding point clouds were also appended to reveal comprehensive results. 3D detections from Point Fusion were projected on the point clouds to generate front view detections. When compared with BEV detection results in Figure 1-3, it became evident that Point Fusion also gives better performance when detections were made in point clouds. This performance improvement was justified by the fact that sensor fusion networks extract features from both 2D images and point clouds exploiting intensity, height, and density information. There is a partially visible vehicle in Fig. 1(a) that was not spotted by YOLOv3.The other two networks, on the other hand, caught the identical car, demonstrating that point clouds and sensor fusion are more capable of recognizing partially visible objects than 2D images.

Moreover, in Fig. 2(a), YOLOv3 missed several occluded objects whereas maximum occluded objects were detected by Point Fusion. This was an important observation as performance in an occluded environment is an important parameter to evaluate the performance of networks. While sensor fusion frameworks are computationally complex and have greater inference time as reported in Table 1, these challenges can be traded off for better performance and accuracy of detection.

Conclusion.

We provide a comparison of three object detection techniques based on the input data type in this paper. An image-only algorithm, a LiDAR-only method, and a sensor fusion framework are among them. The KITTI benchmark dataset is operated to test these object detection systems. Performance evaluation concerning three metrics – i.e., accuracy, performance in occluded environment, and computational complexity – show that the sensor fusion framework gives better overall performance than single sensor algorithms. Qualitative and quantitative results expressed also support the thesis that sensor fusion for object detection is more productive as compared to camera and LiDAR only algorithms. As part of future work, we intend to explore the performance improvements achievement due to sensor fusion in the context of overhead vehicle profiling for Intelligent Transportation Systems (ITS).

Acknowledgement. The manuscript has not been published to other journals and all authors contributed significantly to the research.

Author's Contribution. All authors contributed significantly towards the completion of this research and all authors are in agreement with the content of the manuscript.

Conflict of interest. There exists no conflict of interest for publishing this manuscript in IJIST as the manuscript has not been published or submitted to other journals. However, this research was presented in International Conference on Engineering & Computing 2021.

Project details. This work was supported by the Higher Education Commission of Pakistan (HEC) Technology Development Fund (TDF) grant TDF03-219 awarded to Dr. Shahzor Ahmad.

REFRENCES

1. G. Ross, *"Rich feature hierarchies for accurate object detection and semantic segmentation,"* in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

2. G. Ross, "*Fast R-CNN,*" in Proceedings of the IEEE Conference on Computer Vision (ICCV), 2015.

3 R. Shaoqing, *"Faster R-CNN: Towards real-time object detection with region proposal networks,"* in Proceedings of the IEEE Conference on Neural Information Processing Systems (NIPS), 2017.

4. H. Kaiming, "*Mask R-CNN,*" in Proceedings of the IEEE Conference on Computer Vision (ICCV), 2017.

5. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi*, "You Only Look Once: Unified, Real-Time Object Detection,"*in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

6. J. Redmon and A. Farhadi, *"YOLO9000: Better, Faster, Stronger,"*in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-652.

7. J. Redmon and A. Farhadi*, "YOLOv3: An Incremental Improvement,"* 2018.

8. A. Geiger, P. Lenz, and R. Urtasun*, "Are we ready for autonomous driving?The kitti vision benchmar suite,"* IEEE CVPR, 2012.

9. C. R. Qi, H. Su, K. Mo, and L. J. Guibas, *"PointNet: Deeplearning on Point Sets for 3D Classification and Segmentation,"* in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

10. S. Martin, *"Complex-YOLO: Real-time 3d object detection on point clouds,*" in Proceedings of the European Conference on Computer Vision (ECCV).

11. S. Song, and J. Xiao, *"Deep sliding shapes for amodal 3d object detection in rgb-d images,*" in In Proceedings of the IEEE Conference on Computer Visio and Pattern Recognition, 2016.

12. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia., *"Multi view 3D object detection network for autonomous driving,"* IEEE CVPR, 2017.

13. B. Li, *"3D Fully Convolutional Network for Vehicle Detection in Point Cloud,"* in IROS,, 2016.

14. D. Xu, D. Anguelov, and A. Jain, *"Point Fusion: Deep Sensor Fusion for 3D Bounding Box Estimation,*" in Proceedings of the IEEE Conference on Computer Visio and Pattern Recognition,, 2018.

15. X. Du, M. H. A. Jr, S. Karaman, and D Rus, *"A General Pipeline for 3D Detection of Vehicles,*" IEEE ICRA, 2018.

16. B. Li, T. Zhang, and T. Xia, *"Vehicle Detection from 3d lidar using fully convolutional network,*" In Robotics: Science and Systems, 2016.

17. D. Nister, O. Naroditsky, and J. Bergen, *"Visual Odometry,*" IEEE CVPR, 2004

18. T. Y. Lin, P. Goyal, and G. Ross, *"Focal Loss for Dense Object Detection*," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

19. W. Liu et al., *"SSD: Single Shot MultiBox Detector*," 2016 ECCV